



MONGO'S TOO EASY

Description: An anti-virus system infects its own users. Apple's next iOS release "fuzzes" cellular locations. cURL discontinues bug bounties under bogus AI flood. AI discovers and fixes 15 CVE-worthy zero-days in OpenSSL. Ireland did NOT already pass their spying legislation. AI irreversibly deletes all project files. Says it's sorry. Windows has a serious global clipboard security problem. ISPs have the ability to monetize their subscriber's identities. MongoDB has lowered the hacking skill level bar to the floor.

High quality (64 kbps) mp3 audio file URL: <http://media.GRC.com/sn/SN-1063.mp3>

Quarter size (16 kbps) mp3 audio file URL: <http://media.GRC.com/sn/sn-1063-lq.mp3>

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. He's going to talk about an antivirus that infects its own users. Hmm. That's not good. cURL discontinues bug bounties. That's not good, either. They say they have to do it. And MongoDB has lowered the hacking skill level bar to the floor. It's too easy to hack. All of that and more coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 1063, recorded Tuesday, February 3rd, 2026: Mongo's Too Easy.

It's time for Security Now!. Oh, goodie, goodie, goodie. I don't think all those CISOs and CIOs and security professionals listening are going, oh, goodie, goodie, goodie. But in their heart of hearts, they're thinking, yay, it's Tuesday. Steve's here. Yay.

Steve Gibson: What are they going to talk about today?

Leo: Steve Gibson, our hero, the man of the hour. Every Tuesday we get together, talk about the latest security news. And, you know, interestingly enough, there's never been a lack of security news to talk about.

Steve: Oh, boy. And in fact Lorrie's been pushing me to start working on the podcast earlier in the week. Well, and it makes sense because she knows how stressed I get.

Leo: Oh.

Steve: You know, when I commit to doing something, and doing a good job, that's going to happen. So I was reminiscing with her that there was a time, maybe a couple years

ago, when I would come, you know, because I'm working in my separate location during the day, and then so I would come home at 4:00, and I would say, on a Monday, and I would say, "I have all my topics." Like I've gone through the news, and I've got a list of topics. And then Monday night and all day Tuesday morning, up until we started recording I'd be fleshing everything out, doing the research, pursuing the leads, you know, basically creating the show notes that we now see. And she'd been pushing me to, like, start sooner, start earlier. And so then the other thing that's happening is we're in the process of working on this, finishing up this remodel, which is now 18 months in, I mean, we bought the house...

Leo: Oh, I know all about that.

Steve: ...a year and a half ago. And so then there's the problem that I'm needed onsite for, like, decision-making. And for example this morning - wait, no, no. Yes. This morning it was supposed to be, I told them I'm available until 1:00. Well, then it was going to be noon. Then they switched it to 9:30, and these were people coming out to measure the stairs for the main staircase railing, and I needed to be there. Well, I couldn't be there if I was up against the podcast deadline. So I started this week on Saturday morning. Normally, I try to do coding and GRC stuff all day Saturday. And then she had me starting - she. I mean, you know. Well, I am married, so it is...

Leo: The little woman? Or, as Richard Campbell calls her, "She who must be obeyed?"

Steve: Anyway, yes. And I've heard that, actually that's a common phrase now. And the other one is "happy wife, happy life."

Leo: Happy life.

Steve: So, yeah. Anyway, so the point is that, as a consequence of the fact that I'm starting earlier, and I have to say, it is nice to, like, have it done.

Leo: Yeah.

Steve: Know that I've got my commitment met, and then I'm free to write code. Otherwise I'm sort of preoccupied by, oh, you know, I've got to get to it. So as a consequence, the show notes this week went out, like, Sunday early afternoon. And someone wrote back and said, oh, I love this. This is the earliest I've ever received them. Unfortunately, a super important piece of news dropped.

Leo: That's what came to my mind, yes.

Steve: Yes. And I have been flooded with our listeners because I actually warned about this event for several years. I was saying, eh, this is going to be a problem, or could be. Not - I didn't say it was going to be, but this is a danger. And as a consequence of the fact that I've been, I won't say "predicting it," but recognizing that this is a problem, oh, my god. All of our listeners said, "Steve, it happened."

Leo: Oh, yeah.

Steve: Anyway, so that was about Notepad++ that we'll be talking about.

Leo: Oh, what a story, yeah.

Steve: Yes. And but oh, Leo, there's been a breakthrough. Our Picture of the Week, there is a breakthrough in age verification that does not require its verifier, the person being asked to verify their age - you can't look. Don't look. You can't look yet.

Leo: I save it. I save it. I just see the - I see your headline. That's all I see.

Steve: Finally, an age verification solution that does not require its user to provide any additional information. It's...

Leo: Fantastic.

Steve: It's the kind of thing where it's like, once you see it, it's like, oh. How did we, like, everybody, like, get all tangled up in crypto and everything? It's like, no, no, it's much simpler than that, folks. Anyway, I think maybe this is going to be a good podcast.

Leo: Okay. I'm going to tell you, Steve, we're going to get you vibe coding here. You've really got to get - I know you're a coder, and you probably think, oh, no, code has to be written by humans. But there are certain things like, for instance, I've, over the last couple of weeks, vibe coded a series of tools that I use to prepare the shows now. They go out, I have a news reader that's custom-built just for the kind of news reading that I want to do. You know, you would point it to all the security sources. And not only does it let me bookmark them, but it summarizes them. It pulls quotes and stuff. You could have - and I have a workflow, so I do that.

And then I have a - that's called "beat check," and then there's a tool after beat check that I run every day called "collect stories." It goes out, collects stories for each of the shows, puts them in a format that you can read on the web, and that I can open with Emacs so I can organize it. And then there's a final stage called "prepare briefing" that prepares a web page for it.

You could easily vibe - all of this is vibe. I didn't write a line of code. All of it's - I just said, well, could you do this? Could you do that? Could you upload it? Could you do this? And by the time it's done now it's not only saved me a lot of time, but it's given me a whole, you know, way of doing this that takes a lot of the stress out of it. I think we should talk because I would like for you to try these tools. You will be blown away.

Steve: The difference that I see is that - and I've heard you talking about this on your other podcasts, that you've got, like, too many topics.

Leo: Right, I can't stop.

Steve: You've got too many things to talk about. What our listeners have told me they value is...

Leo: Your decisions, yes.

Steve: ...my analysis of, you know...

Leo: Well, and that step is not gone. This isn't so automated that I'm out of the loop. But it just gets the stuff ready for me. And then I go, not going to do that, not going to do that. Oh, that one's good. That one's good. That one's good. And the AI summaries help with assessing that.

Steve: Yeah, I don't have any dearth of topics, as you said, in security.

Leo: No, no. Nor do I. Right. Yes. No. A lot of what we do is boil it down. That's our job, right, is to take this flood of information and make it usable for our listeners.

Steve: Yeah.

Leo: You do a great job of that. I'm just thinking it's a - well, I'm not going to talk you into it. It's if you want. But I think you - I would be very interested in your reaction, just to see, as this thing writes its code, how competent it's gotten over the last two weeks, three weeks. November 24th was the breakthrough day of last year.

Steve: Wow.

Leo: If we're not at AGI, then I think we need to define AGI better. And this is very human, very competent, very responsive.

Steve: One of our topics - we have two main topics for today. I titled this "Mongo's Too Easy," coming back to...

Leo: Which made me think of "Blazing Saddles." But I don't think that's that Mongo we were talking about.

Steve: Because it was the first podcast of the year was talking about a Mongo breach.

Leo: MongoDB.

Steve: Or Mongo Bleed, rather. Mongo DB.

Leo: Right.

Steve: There's more information about that. But there's another piece of information that was vying - actually it was the topic - it was my working topic until there were two things that I wanted to talk about. The other is the breakthrough in bug finding. It's another thing that we thought was going to happen, but one of the trends I would say it's safe to make, and it echoes what you're saying, is this is happening faster than anyone expected. Right? I mean, it was...

Leo: Mindbogglingly fast.

Steve: We knew it would happen. I had said AI should be able to code because code has a rigor that, you know...

Leo: It's their native language.

Steve: ...psychotherapy doesn't.

Leo: Yes. It's the native language.

Steve: It's like, well, sorry you're not feeling well today, honey; but, you know...

Leo: You know, it's funny, though, one of the things that's come up, and I don't know if you cover it today, but a lot of the things people are doing now with things like OpenClaw is completely violating all of everything we know about security. You know, it's gotten to the point now where cURL into BASH, no big deal. Not going to - I'm not going to - and in order to use these personal assistants, you basically have to say, well, I'm just going to throw caution to the wind. In fact, the new phrase is "YOLO Everything." You only live once. Just YOLO it all. And it's very tempting because when you do, you know, that balance between security and privacy or security and convenience.

Steve: That functionality.

Leo: Functionality.

Steve: Or ease of use.

Leo: The scale is vastly tipped. If you're willing...

Steve: A perfect example is what the shuttle's computer programming cost. Because it could not have a single bug.

Leo: You can't fix it?

Steve: So it was insanely expensive...

Leo: Right, right.

Steve: ...to program that poky little computer.

Leo: Right.

Steve: In the shuttle.

Leo: So at this point the temptation to give this AI agent, this Clawbot, it's called OpenClaw now, they keep changing the name, but OpenClaw, your credit card number, your phone number, access to your Google Docs, your Gmail, everything. The benefit you get out of that is so great that it's very tempting. I'm sitting here, it's half installed. My fingers, it's like I just can't bring myself to do it. What I'm going to do is give it a credit card with a hard limit of like \$5 a day. But you want to give it all these tools because it's amazing.

There was a guy I was watching, looking at a guy on Twitter who said, this is weird, but I told my Clawbot to surprise me, you know, work on something overnight and surprise me. It called him. It made a phone call. It had overnight gotten a phone number, created a computer-generated voice, and called him in the morning and said, "Hey, surprise, I figured out how to make phone calls." Steve, it's getting weird out there. Anyway, we're going to talk about security and our Picture of the Week, and we solved the age-verification problem. It's fantastic.

Steve: Oh, Leo, it is the most brilliant solution.

Leo: Fantastic. We will talk about that in just a moment when Steve - I'm going to shut up now because this is Steve's show, and it's all about Steve. All right.

Steve: Okay.

Leo: Are you being facetious when you say this solves everything?

Steve: Oh. I have to interrupt. I have to preempt. We don't normally do breaking news here.

Leo: Uh-oh. That can't be good.

Steve: But while you were talking, I got a little blurb on my phone, a little piece of news.

Leo: Yes?

Steve: The Wall Street Journal just posted: "AI Disruption Fears Roil Software Industry and the Stock Market." And it says: "From Legalzoom.com and Expedia to Ares and Apollo, shares of companies that sell or invest in software fell sharply on Tuesday."

Leo: Yup.

Steve: "Investors' fears that new developments in artificial intelligence will supplant software reverberated through the stock market Tuesday, dragging down the shares of companies that develop, license, and even invest in code and systems."

Leo: Oh, boy.

Steve: "Traders have questioned."

Leo: I just lost a house worth of money.

Steve: "Traders have questioned whether AI will chip away at the competitive moat built by software makers like Adobe and Salesforce, ever since generative AI models hit the market several years ago. Recent advancements in tools such as those from AI developer Anthropic are now prompting more scrutiny. On Tuesday morning investors homed in on Anthropic's announcement that it was adding new legal tools to its co-work assistant meant to help automate a number of legal drafting and research tasks. Shares of Thompson Reuters, Legalzoom.com, and London Stock Exchange, which all provide some form of legal tools..."

Leo: Very expensive.

Steve: "...[crosstalk] basis fell more than 10%."

Leo: Yup. And I think that's true. But it's disruptive. There will be opportunities. For instance, I think a huge opportunity is enterprise-grade security around these AI tools. The stuff I want to do with OpenClaw is so risky, you would never let somebody do that in a company. Ever. But there will be companies that will come up with ways to do this in a secure and safe fashion. Those guys are going to make a lot of money. So it's...

Steve: But what it means is expanding the security boundary, expanding the moat to encompass much more than it did before.

Leo: Absolutely.

Steve: We had lots of small security boundaries that were all individualized. What we want to do then is to expand that so that there's much more content within a much larger boundary. In that case, then all of that is able to interact within its own enclave.

Leo: And there has to be - there have to be AI firewalls. There have to be ways of letting AI go out and look at the world without exfiltrating your private company documents or your credit card numbers. There's going to be ways to do this, I'm convinced.

Steve: I would argue that that's probably the challenge. We've talked a lot about how, you know, adding an equal sign to the end of a prompt breaks through like all of the protections. It's like, what? So you know, this doesn't work in any normal way that we've known before. But it's also not surprising that that's where the answer was. And no one thought to look there before. Anyway, I never mentioned we're going to talk about an antivirus system which is infecting its own users. Yeah. Whoops.

Leo: Handy.

Steve: Not what you look for in AI.

Leo: Eliminate the middleman.

Steve: Or in AV. Apple's next iOS release, a point release, will be fuzzing cellular locations.

Leo: We've talked about that, yeah.

Steve: cURL has discontinued its bug bounty program due to a flood of bogus AI-generated bug reports. They just said, okay, we can't - no more payout anymore. We have the other main topic I wanted to talk about is AI discovering and fixing - get this - 18 CVE-worthy zero-days in OpenSSL.

Leo: Holy cow.

Steve: This is the breakthrough on that side that we need to talk about.

Leo: Yup.

Steve: It turns out that Ireland, contrary to what I said last week, did not already pass their spying legislation. We have a listener in Ireland who is involved in performing Irish-English translations and explained to me why this was confusing. We'll share that. An AI irreversibly deleted someone's project files and apologized. Yeah. It was very polite about it.

Leo: At least it apologized.

Steve: AI is very polite, Leo. Oh, I'm sorry, you're dead? That's too bad.

Leo: Oh. My bad. I am so sorry.

Steve: We're going to look at Windows' serious global clipboard security problem. Another listener came up with something I hadn't thought of before about a way for ISPs to monetize their subscribers' identities. And then we're going to look at MongoDB having lowered the hacking skill level bar to the floor. So lots of good stuff to talk about. But now, Leo, it is time to share with our listeners a stunning breakthrough in age verification.

Leo: I'm very nervous about this. Let's see. Okay. I'm going to...

Steve: I'll let you read it and react, and then I will explain it.

Leo: I'm going to scroll up. "We need to verify your age. Please choose a verification method below. You only need to do one method." Take a selfie, okay. Search for my ID in existing breaches. Is this real?

Steve: So you have a multiple choice here. This is for age verification. You know, the headline says "We need to verify your age. Please choose a verification method below." You've got two choices here. And it does give you a little padlock and shows "Your details are used for verification purposes only and is not stored." And so the first choice is take a selfie. "Confirm your age with a quick selfie, which is processed directly on your device for privacy." Or you could choose the second option, which is "Search for my ID in existing breaches." And it explains: "We'll search for your ID in our database of breached personal information. If your ID is found, we can verify your age automatically. It's quick and easy, and odds are you're already in there."

Leo: Oh, lord. That's true. Now, is this a joke, or is this serious?

Steve: No. It was...

Leo: It could be serious. I mean, that's true; right?

Steve: Yes. It's one of those things where you have to do a double-take because you're thinking, uh, wait a minute. And it's powered by KidID. That's the logo at the bottom. And one of our listeners who received this Sunday afternoon wrote and said, "Wait a minute, KidID.com is a real thing."

Leo: It's real, yeah.

Steve: It's like, yes, they are in fact the service that, god, now I don't - one of the people that we've talked about, one of the age clamped services was using KidID in order to perform this verification. And I think maybe they're the ones that were not deleting people's selfie pictures and got caught doing that.

Leo: Oh, mm-hmm.

Steve: But I could be wrong about that.

Leo: But good news. Now they have my ID in their breach, and I'm set.

Steve: Yeah. So basically, security breaches are - so I'm sure this was a fake.

Leo: Yeah, it's got to be tongue-in-cheek.

Steve: But it's, you know, great humor, and it just suggests that, like, well, you know, breaches are so rampant that why are we even being asked to identify ourselves?

Leo: Everybody knows how old you are.

Steve: That's right.

Leo: Wow.

Steve: That's exactly right. Okay. So I'm going to do this out of order because I want to address this big piece of news that really lit up our listeners. Don Ho, the author of the immensely popular Windows Notepad replacement, which is Notepad++, which I, along with many of our listeners, have chosen to use for, you know, as like our primary simple text editor. And, I mean, ++ is no exaggeration. This thing, it recognizes the language of what it is you're dropping into it based on file extension. It's got every bell and whistle you could imagine. So, I mean, and over time I've really come to like it.

Well, Don notified the world Sunday, after the podcast notes went out, that for around six months or so, like through the second half of last year, June through the start of December 2025, unbeknownst to him, of course, highly sophisticated state-level actors believed to be Chinese had arranged to compromise, and did compromise, his Notepad++ software update mechanism. They used it to launch targeted malware attacks against specific Notepad++ users. So a, you know, a serious supply chain attack. Now, our listeners know that I have complained on multiple times about the high rate of Notepad++ updates. Bless his heart, Don seems unable to just leave this thing alone. It's, like, it's never done.

So I specifically cited the possibility of exactly this sort of supply chain attack being facilitated due to Notepad++ seemingly endless code changes. Every time it downloads another copy into your computer, that's another opportunity. I'm not saying that it's going to happen, but it could. You know? And the more frequently it's done, of course what happened is these Chinese state-level bad guys, they're not dumb. They see that

Notepad++ is updating itself, like, not hourly, but, like, all the time. And they're thinking, hey, that's a target. But, you know, we want to get that because it's abused all of its users into accepting constant updates. And every update is another opportunity for us to get our malicious code into someone's computer. And that's what happened.

So anyway, I wanted to acknowledge to everybody that I got everyone's emails. Thank you. I'm glad, you know, everyone says, "I'm sure you already know about this, but...." And the first time I get one of those, that's not true. But all subsequent ones of course it is true. Still, I do appreciate them, and I appreciate having everyone make sure that I knew about this. And also Don acknowledges, and if it was important, and it's not because the problem's been solved already, but he also had a lack of security in his own update mechanism, which his compromised hosting provider - this was not a compromise at his end. It was the system which was hosting his updates is what got compromised. But they targeted him and his Notepad++.

So it's now at 8.9.1. He recommends that you go to the site, download it yourself, and perform a manual installation, just to be absolutely sure. And I would say then turn off this whole automatic update nonsense. We have one listener who was proudly strutting around saying, I'm at 8.2 or something from three years ago, and I'm sure glad I turned it off back then. So again, and I mean, these things are like, oh, the accent on the Swedish umlaut is backwards. It's like, oh, so let's update the world with a new copy. It's like, no. Come on. Just, you know. It works. And you know, Leo, one of the things I've always appreciated about firmware updates is the manufacturer recognizes that a firmware update is, you know, it's a little bit fraught. Right? If you trip over the power cord in the middle of a motherboard firmware update...

Leo: Right.

Steve: You don't have a motherboard anymore. So their advice is always, if everything is working, don't update your firmware because, guess what, everything is working. It's only if you've got some known problem that a firmware update is known to fix that it makes sense for you to, you know, make sure the plug is tightly in the socket in the wall, and keep the dog, you know, in the other room, and then start your update. So again, Notepad++ has been fine for like the last decade.

Leo: They think it's Chinese, as you said, the Chinese hackers. What do you think they were after? Just to get on as many machines as they possibly could?

Steve: No, no, no. I want to believe everything that we're being told. So it's a huge relief that these were apparently very targeted. They were looking to get into specific machines, and they did, using Notepad++ as their trojan to get them onto the machine. But so none of us, none of our, I mean...

Leo: So it wasn't a crypto stealer or something like that.

Steve: Correct.

Leo: It was really aimed at probably Chinese dissidents or overseas Chinese, something like that.

Steve: Yes. Yes. And I did see that the attacks that were known were over - they were targeted at other Asians over there, not aimed at the West. I'll just note, though, I mean, I've downloaded Notepad++ updates sometime between last June and the beginning of December. Although had this been a widespread attack, it would have come to light much quicker. So certainly the reason we believe these were high-level Chinese state actors is they didn't want this to get found. They wanted to keep this facility of being able to selectively infect specific Notepad++ users alive and working for them, available to them as long as they could.

So it's a good thing that it wasn't a widespread attack because anybody updating during that window of the time the attack starts and it is found and ended would have had malware installed in their computer. And, you know, many of us are updating Notepad++ a lot. I also stopped back when I said I'm sick and tired of this, and I turned that off. So, but I don't know that I haven't done it since last June. So anyway, really, if it's, you know, we're like all addicted to this update, update, update. We've got to have the latest and greatest because maybe it's going to fix some big problem that we don't know we have. Well, if you don't know you have it, you're probably okay.

So last week, two security companies - Morphisec and Kaspersky - both detected and reported that the eScan antivirus product, published by a company based in India, had attacked its own users after one of its - get this - update servers was breached and infected with malware. So this perfectly reflects what we were just talking about with Notepad++. You know, we are seeing an increasing incidence of supply chain attacks. And attacking people's insecure update servers because, as I said, the world's become addicted to updates. Everything we've got is updating itself all the time. So here again is another instance of that. The event was covered by BleepingComputer, which shared eScan's defensive annoyance over the bad press this generated. And I'm here to give them some more bad press because oopsie.

BleepingComputer also reminded us that back in April of 2024, so coming up on two years ago, eScan's update facility was breached by North Korean hackers and used to spread malware into corporate networks. So, you know, I've often said that anyone can make a mistake. It's true. And that sometimes mistakes make us stronger. But an antivirus solution has a very, a highly privileged position in our machines. It's got to be running in the kernel. And a second similar incident occurring fewer than two years after the first one, I think that should be a concern to any eScan customer. That's a reason to look elsewhere for an antiviral solution, if you want to look anywhere at all. So of all the coverage...

Leo: You don't really - no one, you don't, I mean, I guess a business might but...

Steve: I'm going to get there. I'm going to get us there here in a second.

Leo: Okay.

Steve: Because I completely agree with you.

Leo: Yeah.

Steve: Of all the coverage this received I thought that Kaspersky summarized the technical details best. They explained: "On January 20th" - so, right, a couple weeks ago

- "a supply chain attack has occurred, with the infected software being the eScan antivirus developed by an Indian company, MicroWorld Technologies. The previously unknown malware was distributed through the eScan update server. The same day, our security solutions detected and prevented cyberattacks involving this malware. On January 21st" - meaning a day later - "having been informed by Morphisec, the developers of eScan contained the security incident related to the attack.

"Users of the eScan security product received a malicious Reload.exe file, which initiated a multistage infection chain. According to colleagues at Morphisec who were the first to investigate the attack, Reload.exe prevented further antivirus product updates" - of course it would - "by modifying the hosts file, thereby blocking the ability of security solution developers to automatically fix the problem which, among other things, led to an update service error."

Okay, now, I want to take a moment here just to remind everyone how very powerful the hosts file remains and to share a little bit of Internet historical trivia. The presence of a hosts file predates the Internet. As we know, ARPA stands for the Advanced Research Projects Agency, and the Internet grew out of the earlier work on something that was known as ARPANET. I recall that when I was working at SAIL, Stanford University's Artificial Intelligence Lab, in 1972, a big refrigerator-like thing, white, and looked like it came from a battleship, I mean, it was really overbuilt. It was just then, while I was there, being installed. It was an IMP (Interface Message Processor) which was a node on the still very young ARPANET.

Back before the creation of DNS there was a need to map familiar host names to ARPANET addresses, or nodes. And as we know, that's the role that DNS serves us today. But ARPANET had no DNS. It barely even had ARPANET. So every machine on the ARPANET had a copy of the ARPANET's master hosts file. That file was maintained on a single machine at SRI (Stanford Research Institute), and all hosts on ARPANET would periodically pull that file from SRI's one designated master copy to maintain an updated and synchronized listing, a view of all other available machines on ARPANET.

Leo: We have in our Discord right now a guy who worked at Bolt, Beranek and Newman, BBN...

Steve: BBN, yup.

Leo: Who says I actually drew the ARPANET apps when I worked at BBN, before there was anything called DNS.

Steve: And before we had CAD. So he was drawing them with...

Leo: By hand, yeah, with a protractor.

Steve: A stencil and then, you know, and a straight edge, yeah.

Leo: Yeah. Very amazing, Craig. Wow.

Steve: So in a classic example of old computer stuff sticking around from generation to generation, the original hosts file never went away. Today, it sits somewhere inside every Internet-connected machine. Windows users can find it at C:\Windows\System32\drivers\etc. So, I mean, it's like really an afterthought; right? Drivers\etc.

Leo: Etcetera.

Steve: I just looked at mine on my Windows 10 machine. Its first line of that file contains a Microsoft copyright notice dated 1993. So like when the TCP/IP stack was first added to Windows 95 because the file was dated '93; right? Or maybe Windows 3.1. I don't remember what the first Windows was that got on the Internet. Anyway, the thing that makes the hosts file so powerful is that, by convention, it is the first place any Internet-connected machine will look for a host name to IP address mapping. In other words, it takes priority over everything else. And you don't even have to restart or reboot. I've used this sometimes myself when I've needed to locally test some client-server code that will eventually run at www.grc.com.

If I add the line 127.0.0.1, space or tab, you know, some form of white space, then www.grc.com, then immediately and without waiting, restarting, rebooting or anything, any attempt to access www.grc.com will be intercepted and be handled by a server on my own local machine. And that allows me to use a TLS, www.grc.com certificate on my local machine. I mean, it's exactly as if it were at GRC.com because the browser thinks that's the domain that it's accessing, and so the certificate works. So anyway, modification to the hosts file can also obviously have malicious consequences. If, as in this case, somebody wished to prevent future updates to eScan's antivirus system, after they'd infected the machine, placing the domain names of those update services into the user's local hosts file would immediately and completely prevent the compromised antivirus from being updated again to eliminate the malware.

Kaspersky continues, writing: "The malware also ensured its persistence in the system, communicated with control servers, and downloaded additional malicious payloads." In other words, you do not want this thing getting into your system. Reload.exe, you know, yeah, reload the gun. "Persistence was achieved," they wrote, "by creating scheduled tasks. One example of such a malicious task is named CoreIDefrag." Oh. Sounds simple. You know, harmless CoreIDefrag. Doesn't make any sense, really, but okay. "Additionally, the constlx.exe malicious file was written to the disk during the infection." Okay. "At the request of the BleepingComputer information portal, eScan developers explained that the attackers managed" - oh, so this is Kaspersky. It's why they referred to BleepingComputer so oddly.

Kaspersky is writing this, saying: "At the request of the BleepingComputer information portal, eScan developers explained that the attackers managed to gain access to one of the regional update servers and deploy a malicious file, which was automatically delivered to customers. They emphasize that this is not a vulnerability." Uh-huh. "The incident is classified as unauthorized access to infrastructure." Right.

Leo: Yeah.

Steve: We're not going to call it a vulnerability. Right. Even though all of our customers got infected. "The malicious file was distributed with a fake invalid digital signature." Now, that's interesting. Somebody was asleep at the switch and didn't notice that the signature, the digital signature was invalid, or didn't stop this thing from executing. According to the developers, the infrastructure affected by the incident was quickly

isolated, thanks to other people finding it and telling them, and all access credentials were reset.

"Having checked our telemetry," writes Kaspersky, "we identified" - get this - "hundreds of machines belonging to both individuals and organizations, which encountered infection attempts with payloads related to the eScan supply chain attack. These machines have been mostly located in South Asia, primarily in India, Bangladesh, Sri Lanka, and the Philippines."

Okay, now, I'll take a moment here to note that these are only the "hundreds" of machines that also happen to be under the observation of Kaspersky's telemetry. This must reflect only a tiny microcosm of the entire Internet. One of the things that annoyed me was seeing the MicroWorld Technologies people, because there were other things that I pursued in getting to the bottom of this, they were dramatically pushing back and downplaying the severity of this problem for their customers, which was pretty severe. The one thing we don't want to see is an irresponsible provider of highly privileged antivirus software. You need to trust your AV company.

Kaspersky says: "Having examined them, we identified that to orchestrate the infection, attackers have been able to replace a legitimate component of the eScan antivirus, located under the path C:\Program Files (x86)\escan\reload.exe, with a malicious executable." So that reload.exe in the eScan subdirectory is the problem. They said: "This Reload.exe file is launched at runtime by components of the eScan antivirus. It has a fake, invalid digital signature. We found this implant to be heavily obfuscated with constant unfolding and indirect branching, which made its analysis quite tedious."

Okay, now, what Kaspersky means when they refer to "constant unfolding and indirect branching" is that typical straightforward code simply contains "jump" instructions which cause the program's execution to "jump" to another location. So someone examining a disassembly of the code can see for themselves where the CPU's execution will jump to.

By comparison, an indirect jump refers to another location in the program or to the contents of a CPU register, and it will be the current contents of that location or register that specifies the location to which the CPU's execution will jump. Since there's no way to know what that location or register might contain at the moment the indirect jump is executed, a static disassembly and an examination of the deliberately obfuscated malicious code will not reveal its execution paths. You won't be able to tell by looking at the code itself where anything is going to jump to because you don't know until you actually run the program that those addresses get resolved. So as Kaspersky noted, this makes an analysis of the code far more tedious. And that's of course exactly what its malicious creators intended.

Kaspersky continues, saying: "When started, this Reload.exe file checks whether it's launched from the Program Files folder, and exits if not. It further initializes the Common Language Runtime environment inside its process, which it uses to load a small .NET executable in memory. This executable is based on the UnmanagedPowerShell tool, which allows it to execute PowerShell code in any process. Attackers have modified the source code of this project by adding an AMSI bypass capability to it, and used it to execute a malicious PowerShell script inside the Reload.exe process."

Okay. Now AMSI is Microsoft Anti-Malware Scan Interface. So this malware has arranged to bypass that. I wish my own code did that. Maybe I wouldn't have so many problems...

Leo: Yeah, really.

Steve: ...with Microsoft's annoying antimalware scan.

Leo: Who needs to sign stuff?

Steve: Which is, you know, false positivng on me. Anyway, Kaspersky's teardown goes on to take the malware apart and describe its operation in great detail. But we all have a good sense now for what happened. And the point you were going to make, Leo, I have in the show notes.

I wrote: "Neither Leo nor I use any third-party antimalware add-on; and whenever I'm asked, I recommend against it." It's true. There was once a time when I strongly recommended the addition of a third-party firewall to Windows. Then Microsoft added one into XP and finally set it running by default with XP's Service Pack 3. The same thing happened with add-on antivirus. The various third-party AV solutions had their day, but that day has passed. Windows now brings its own along. I see no benefit, and only downside risk, associated with gratuitously adding another to Windows. This recent misadventure with eScan shows how much trust any third-party must be given to obtain such an honored place in our PCs. As I said, AV is in the kernel. Which means, if it goes bad, you're in deep trouble.

Leo: It goes bad.

Steve: Yeah. It's just not worth it.

Leo: Yeah.

Steve: So the Apple iOS world has been moving through a number of "point" and "point-point" releases. Seems like we've had a lot of updates after 26, yeah.

Leo: There have been, yeah.

Steve: And of course some of that has been good. They've toned down Liquid Glass, making it a little less liquid-y.

Leo: Whew, yes.

Steve: Yeah. Every so often, even with all my settings set to mute it and suppress it, like I'll get a little weird liquid-y squiggle under something.

Leo: Terrible. It's terrible. Why do these companies do this? I don't get it.

Steve: Yeah, yeah. Okay. So we're currently hovering at 26.2.1. But there is some welcome news about 26.3 for cellular-connected devices. Last week Apple announced that it would be adding optional deliberate imprecision to cellular services' ability to geolocate cellular devices. So here's what we learned from Apple under their headline

"Limit precise location from cellular networks." They said: "With the 'limit precise location' setting, you can limit some information that cellular networks may use to determine your location. Available on compatible iPhone and iPad models with supported carriers." Obviously cellular models. "Cellular networks can determine your location based on which cell towers your device connects to." And of course we know also relative signal strength factors in.

Leo: And I learned just recently they can also request GPS coordinates. Did you know that?

Steve: Wow.

Leo: Yeah. I had no idea.

Steve: Over the cell network.

Leo: Yes.

Steve: Does make sense. Lorrie and I are both Verizon subscribers. And in our area it's like this well-known Verizon dead zone.

Leo: Yeah.

Steve: So one of the first things we did when we set up shop there was we got a femtocell, as they used to be called.

Leo: Smart.

Steve: And, you know, you just connect it into your LAN, and now we have five bars, where we used to, like, not even have one sometimes.

Leo: Irks me a little bit because you're using your Internet for their connectivity. But it's the only way you can get on.

Steve: What's worse, there's no way to lock it to your phones. You're providing cell service to your neighbors.

Leo: I didn't know that. Yeah, we used to have to have a femtocell at the old TWiT studios because it was a dead zone for T-Mobile, yeah.

Steve: Anyway, so the point of this is that one of the things in this is a GPS. You have to put a little antenna out. And it, like, it takes a long time for this thing to boot up because it needs to determine for whatever reason its exact location...

Leo: It's building the almanac, yeah.

Steve: ...in three space so it knows, you know, where it is. So I guess I'm not surprised that they're able to ping your phone and say give me your current GPS location.

Leo: And it is a privacy concern because they sell that to - they don't even sell it. They sell it for cheap if they sell it, to law enforcement.

Steve: Oh, it's exact, too, yeah.

Leo: Yeah, it's exact, yeah. [Crosstalk].

Steve: So Apple said: "Cellular networks can determine your location based on which cell towers your device connects to. The 'limit precise location' setting enhances your location privacy by reducing the precision of location data available to cellular networks. With this setting turned on, some information made available to cellular networks is limited. As a result, they might be able to determine only a less precise location, for example, the neighborhood where your device is located, rather than a more precise location" - oh, look, he's in the bathroom right now - you know, such as a street address. "The setting doesn't impact signal quality or user experience," they said.

And they finish, saying: "The 'limit precise location' setting does not impact the" - and this is important - "the precision of the location data that is shared with emergency responders during an emergency call." So again, they also took the time to think this through. "This setting affects only the location data available to cellular networks. It does not impact the location data you share with apps through Location Services." So, you know within the family, and within your community of devices, and where you've said yes, let Google Maps know where I am when I'm using them, that still remains high precision. So he said: "For example, it has no impact on sharing your location with friends and family with Find My." And so forth.

So, okay. At the moment iOS 26.3 is in its third beta pre-release, so it's expected shortly. Once it's available, the setting can be found under the phone's Cellular Data Options, which I thought was not where I would have looked, but okay, Cellular Data Options. And they said that a device reboot may be required in order to change that setting. So it's probably, you know, down in the base band system that's part of the core infrastructure of their cellular technology. And, you know, as I said before, and I know you are, too, Leo, we're annoyed by Apple's constant commercial upselling of their services. It just feels to me like they don't need to do that.

But the flipside is there is no company that I trust more to have my back. Apple has demonstrated their steadfast commitment to their users' privacy over and over through the years. Now, I fully realize that it might really amount to not that much, right, because tracking and privacy invasions are happening well outside of Apple's sphere of control. So there's not a lot they can do overall. But knowing that my handset is arguably doing everything it can to have my back is better than nothing, and it's what I would choose even while, like Leo, you know, neither of us spend that much time worrying about privacy in the abstract. It feels like, you know, good luck.

Leo: There are a couple of footnotes to this. One is, at least according to some sources, this is because Apple now designs its own modems. So they have that C1X, they can do that. The other thing, though, is, and you read it, but maybe you kind of skimmed over it, it says "participating cellular carriers."

Steve: Yes.

Leo: The carrier has to agree to it.

Steve: Oh. Oh.

Leo: And currently in the United States, the only carrier that's agreed to it in the United States so far is Boost Mobile. And there's even some speculation the carriers might actually sue Apple over this, just as, you know, some companies have sued over app tracking transparency because they make money on selling your location. And they're going to say, you know, their excuse will be, oh, no, no, it's how we improve our service. We need to know...

Steve: We need to know exactly where the phone is in order to map the signal strength reception and blah blah blah.

Leo: Right, exactly. It's for your benefit.

Steve: Wow.

Leo: So it remains to be seen how many companies will allow this. I'd be very curious. On the other hand, there may be consumer demand that tells T-Mobile and Verizon and AT&T, you know, you'd better do this.

Steve: Okay. I hate to do another break except this is going to be a long piece.

Leo: No, you don't.

Steve: This is the other big, the big story. It's another breakthrough in AI. So we're at the top of the hour. Let's take a break.

Leo: Let's do it.

Steve: And then we won't have to break in the middle of this.

Leo: I don't hate when you take a break. I just want you to know. I like it. I like it that you have to take breaks, to be honest. Back to you, Steve.

Steve: So I first encountered this next piece of news thanks to a listener, Elardus Erasmus. He wrote: "Hi, Steve. You may have seen this already. I work for a company that makes use of OpenSSL for cryptographic primitives. I evaluate the vulnerabilities as and when they are disclosed to determine the impact, if any, on our products. Just this Tuesday" - meaning last Tuesday - "OpenSSL released new versions fixing 12 previously unknown security vulnerabilities. This is way more than the usual one or two fixes found in a typical OpenSSL security release."

Okay, now, I want to pause to note that the idea that Elardus works for a company that uses and relies upon OpenSSL's cryptographic primitives, and therefore carefully follows, tracks, and examines the consequences of any newly disclosed vulnerabilities which might have, you know, an effect upon their use that just does my heart good. It is so smart; and it's a perfect demonstration of the responsible way to use any sort of third-party library. You know, most organizations would, and do, simply link to the library and never give it another thought. We don't know who he works for; but whoever it is, they understand what I call "non-finger-pointing security."

You know, deflecting responsibility after a breach occurs due to the use of somebody else's vulnerable library might feel good. You know, you get to say it was not our fault, you know. But the breach still occurred. And it occurred to your systems as a consequence of using a library that you weren't being responsible for its use of. You know what I mean. So anyway, I just wanted to take a moment to say that that is just the right way to do this.

In any event, he explains his reason for writing, saying: To my astonishment, all 12 of the newly discovered OpenSSL zero-day vulnerabilities were found by an AI-based cybersecurity company called AISLE. And I don't know what, if it's an acronym, but it's A-I-S-L-E; right? So that's where their name came from, AISLE, A-I-S-L-E. He says: "Here's a link to a blog post from one of their researchers, in case you're interested." What was also interesting from that blog was to learn that AI slop led to the cancellation of the cURL bug bounty program. He finishes: "Thanks for all you do. Best, Elardus Erasmus."

Okay. So the AI-driven security company we learn of here, as I said, is called AISLE, A-I-S-L-E. And the contents of this blog posting that he linked to by one of their AI security researchers, as I said at the top of the show, it was runner-up for today's topic, and you'll quickly see why.

The researcher begins his posting with a TL;DR which reads: "OpenSSL is among the most scrutinized and audited cryptographic libraries on the planet. It underpins the encryption for most of the Internet. They just announced 12 new zero-day vulnerabilities, meaning previously unknown to the maintainers at time of disclosure. We at AISLE discovered all 12 using our AI system. This is a historically unusual count, and the first real-world demonstration of AI-based cybersecurity at this scale. Meanwhile, cURL just cancelled its bug bounty program due to a flood of AI-generated spam, even as we reported five new genuine CVEs to them. AI is simultaneously collapsing the median" - and he has in double quotes "slop" - "and raising the ceiling," meaning real zero-days in critical infrastructure.

Okay. So let's pause here and first take a look at the problem that the cURL project has had. The project's Bug Bounty page, which is at curl.se/docs/bugbounty.html, it was updated with a very short notice, which just says: "Up until the end of January 2026" - which, okay, here we're on February 3rd today; right? So three days ago. "Up until the end of January 2026, there was a cURL bug bounty. It is no more. The cURL project does not offer any rewards for reported bugs or vulnerabilities." Period.

They said: "We also do not aid security researchers to get such rewards for cURL problems from other sources, either." Meaning, you know, you can't go to HackerOne or one of the other bug bounty programs and say, hey, I found a bug. They're out of that game now. "A bug bounty gives people," they wrote, "too strong incentives," as in incentives which are too strong, T-O-O strong incentives, "to find and make up 'problems' in bad faith that cause overload and abuse. We still appreciate and value valid vulnerability reports."

Okay. So now to give this page - that's all they said on the new bug bounty page is basically "ain't none, we're done," because we were, you know, offering to pay people incentivizes them to just make stuff up. And apparently AI is the cause. Okay. So to give it a little more context, I used the Wayback Machine to capture the same page six weeks ago, on December 18th. Before the closure of all cURL bug bounties, the page said, same page: "The cURL project runs a bug bounty program in association with HackerOne and the Internet Bug Bounty. How does it work? Start out by posting your suspected security vulnerability directly to cURL's HackerOne program. After you've reported a security issue, it has been deemed credible, and a patch and advisory has been made public, you may be eligible for a bounty from this program. See the Security Process document for how we work with security issues.

"What are the reward amounts? The cURL project offers monetary compensation for reported and published security vulnerabilities. The amount of money that is rewarded depends on how serious the flaw is determined to be. Since 2021, the Bug Bounty is managed in association with the Internet Bug Bounty who set the reward amounts. If they set amounts that are way lower than we can accept, the cURL project intends to 'top-up' awards. In 2025, typical 'Medium' rated vulnerabilities are being rewarded \$2,500 U.S. each.' So finally," they finish, "who is eligible for a reward? Everyone and anyone who reports a security problem in a released cURL version that has not already been reported can ask for a bounty." And those days are over.

So, you know, when crimes are being investigated, the classic three requirements are "means, motive, and opportunity." You know, could they do it? Why would they do it? And were they in a position to do it? One of this podcast's foundational observations, which followed the explosion and endurance in high-end advanced intrusions, you know, with ransomware and extortion, has been that the thing the bad guys want, the only thing the bad guys want, is our money. They, I mean, much as our personal details are important to us, they could not possibly care any less about the health records, the dating habits, the sexual proclivities, or Social Security numbers of anyone else. They just don't care. The only value any of that has is for extorting those who somehow allowed that data to escape or to become encrypted and thus unavailable to them under an unknown encryption key.

I'm reminding everyone of this fundamental observation because the presence of a vital and vibrant bug bounty system, which rewards, with money, those who discover and responsibly report security vulnerabilities, represents another source of revenue which can be readily abused. We know how crucial cURL's security is. Leo was just making a joke about, you know, using cURL to BASH [crosstalk]. Oh, well, who cares? Hope for the best.

Leo: You only live once.

Steve: Hope for the best. That's right. We've covered the discovery and remediation of previous critical vulnerabilities in cURL. We also know the importance, the necessity, of motivating security researchers to go looking for problems. Independent researchers need to eat, too. So they're far more likely to look for, discover, and report security

vulnerabilities in open source projects that will reward their time and trouble, than those that do not. cURL's announced withdrawal from their historical and important bug bounty programs means that independent research into cURL's security has effectively ended. You know, sure. You can find one by mistake and report it to them. But sorry, you're just, you know, you're a good citizen. They're not paying anymore.

So I dug around a bit for some additional background, and I found some over at the "It's FOSS" site. The posting, titled "cURL Gets Rid of Its Bug Bounty Program Over AI Slop Overrun," provides some additional background. The guy there wrote: "Last year in May, the cURL project's bug bounty program was inundated with AI slop, where many bogus reports were opened on HackerOne, leaving the cURL maintainers to go through garbage. The problem didn't stop even after Daniel Stenberg, the creator of cURL, threatened to ban anyone whose bug report was found to be AI slop."

He said: "We're now in 2026, and the situation has reached a tipping point. For context, cURL is an open source command-line tool used by billions of devices worldwide. Daniel has submitted a pull request on GitHub that removes all mentions of a bug bounty program from cURL's documentation and website. Coinciding with that, the project's security.txt file has been updated with some blunt language that makes the new policy crystal clear."

Okay, now, we've talked about these types of files previously. They're a semi-formal collection of files that can be found under the /.well-known/directory in the root of websites that have them. So I checked out the cURL project's security.txt file, which reads: "Project cURL. The cURL open source project accepts security reports for problems found in products made by the cURL project. We offer NO [that's in caps], (ZERO) [in parens] rewards or other kinds of compensation for reported problems. But we offer gratitude and acknowledgments clearly stated in documentation around confirmed issues. We will ban you and ridicule you in public if you waste our time on crap reports."

So it does appear that the cURL project is pretty fed up with the nonsense they've been subjected to for the past eight months or so. The posting over on the FOSS continues, saying: "The cURL team intends to make a proper announcement in the coming days, though many outlets have already covered the news of this happening, so I would say they ought to get on it ASAP. The program officially ends in a few days on January 31st, 2026. After that, security researchers can still report issues through GitHub or the project's mailing list, but there won't be any cash involved."

"What pushed them over the edge, you ask? Well, just weeks into 2026, seven HackerOne reports came in within a 16-hour period in just one week. Some were actual bugs, but none of them were security vulnerabilities. By the time Daniel posted his recent weekly report, they'd already dealt with 20 submissions in 2026.

"The main goal here is said to be stopping the flood of garbage reports. By eliminating the monetary incentive, they are hoping people (or bots?) will stop wasting the security team's time with half-baked, unresearched submissions. He also gives a stern warning to wannabe AI sloppers, saying that: "This is a balance, of course; but I also continue to believe that exposing, discussing and ridiculing the ones who waste our time is one of the better ways to get the message through: You should NEVER [all caps] report a bug or a vulnerability unless you actually understand it - and can reproduce it. If you report anyway, I believe I am in the right to make fun of - and be angry at - the person doing it.

"So, yeah," he says, "that's that. If people still don't understand that AI slop is harmful to such sensitive pieces of software, then sure, they can go ahead and make a fool of themselves."

Okay. So that's the bad news. It appears to be a new problem created by AI that will be the automation of the generation of low-quality, often bogus, security bug reports, on the hope that they may score one and get some money. This has the potential to significantly spam the industry's critical bug bounty system. We know that the bounty programs, whose importance has been well established, won't go down without a fight. So what's likely to happen will be much more focus upon the establishment of any would-be bug bounty recipient's reputation. The result would be that reports coming in from unknown, presumably AI bots, hoping to score a bounty, would somehow be treated differently. The problem is, then, it's unclear how an unknown human researcher would go about establishing a reputation as a non-bot. Maybe just submit one high-quality report and wait for it to be seen to be such and, you know, get a gold star, and you've got to get a couple until your reports are less filtered. So anyway, this will be something for us to all keep an eye on.

Leo: I have kind of a little different take on it.

Steve: Good.

Leo: First of all, I don't know this guy. But open source maintainers for very good reasons are cranky as hell.

Steve: Persnickety?

Leo: And he's probably been doing cURL without compensation, without much credit, one of the most used programs in the world for years and is, you know, a little sensitive. I get at least two or three bug reports on our website every day. Not AI generated. Just by people who are hoping to get some money out of us. We don't have a bug bounty even. This is a problem, a people problem, not an AI problem. There are lots of people out there who, you know, are hoping to get some money from somebody by saying I found - you don't get any of these? Because we get them all the time. You know, trust and safety or @twit.tv or that kind of thing, saying you've got a bug, and I'll reveal it if you give me some money. That's just a people problem. Maybe AI has enabled some of these people.

But I think that that's not exactly really the target. And I think there's a huge risk at stopping his bug bounty because there's a lot of people who do make money at this, who legitimately report bugs, who will not be incented to do so. And I think cURL's a pretty important thing. I think the solution to this is not to turn the bug bounty off, but to get some help, to get some more people working on this project and maybe some more eyes on the reports.

Steve: Yeah. I think...

Leo: The final thing is I don't think ridicule's going to do anything because the people who do this are not susceptible to ridicule.

Steve: You'll never hear me ridicule anybody ever. That's just not [crosstalk].

Leo: Well, they're anonymous, for the most part. You know, these aren't real security researchers. I think it'd be fairly easy to filter out these bad reports. I certainly pay no attention to the emails I get every day saying there's a bug on your website.

Steve: Yeah. I think that, for me, establishing a reputation system. We know we need a bug bounty program. We know that bounties are good.

Leo: That's the thing. We shouldn't throw the baby out with the bathwater here. We need that bug bounty. It's a much better solution.

Steve: And sadly, cURL, as you said, I mean, it is on the front lines. We've covered some serious...

Leo: There have been a lot of bugs. That's maybe the other reason he's a little prickly, is if you look at his CVEs, it's not the most secure software ever.

Steve: No.

Leo: I think...

Steve: It's not quiet.

Leo: He might be a little sensitive, at this point, to people, you know, finding bugs? I don't know.

Steve: Yeah, and you know...

Leo: I love cURL. I'm grateful to it. I would contribute to cURL. Absolutely, yeah.

Steve: Stepping back further, too, we know that there is a fundamental problem with the open source model.

Leo: That's right. That's really the problem.

Steve: I mean, you know, major corporations are taking advantage of open source and, you know, and that fantastic cartoon of the whole Internet, you know, resting on a little peg that's supported by someone in Nebraska. I mean, it is a weird system that we've evolved, where one unpaid volunteer is expected to maintain a command line tool used by billions of systems.

Leo: Right. Right. So I'm very sympathetic. More people should support him. The work is very important. Get some help. But I don't think turning off the bug bounty is really - and ridicule is absolutely useless.

Steve: Yeah. Okay. So that's - it's about time for good news, Leo.

Leo: Yeah. Because there's another side of this story; isn't there.

Steve: Yes. That was the bad news. It also appears to be the case that code-digesting and understanding AI, when in the hands of actual security researchers, can create newfound leverage enabling the high-fidelity discovery of true security vulnerabilities.

The first line of the posting by AISLE's security researcher said: "OpenSSL is among the most scrutinized and audited cryptographic libraries on the planet." We know that is not hyperbole. It's absolutely true. I mean, it is really rare to find a bad problem in OpenSSL because the entire industry is being so careful with it. Unlike Daniel with cURL. So here's what this guy went on to explain.

He said: "We at AISLE have been building an automated AI system for deep cybersecurity discovery and remediation, sometimes operating in bug bounties under the pseudonym Giant Anteater. Our goal was to turn what used to be an elite, artisanal hacker craft into a repeatable industrial process. We do this to secure the software infrastructure of human civilization before strong AI systems become ubiquitous. Prosaically, we want to make sure we don't get hacked into oblivion the moment they come online.

"No reliable cybersecurity benchmark reaching the desired performance level exists yet. We therefore decided to test the performance of our AI system against live targets. The clear benefit of this is that for a new, zero-day security vulnerability to be accepted as meriting a CVE, it has to pass an extremely stringent judgment by the long-term maintainers and security team of the project, who are working under many incentives not to do so. Beyond just finding bugs, the issue must fit within the project's security posture, i.e., what they consider important enough to warrant a CVE. OpenSSL is famously conservative here. Many reported issues are fixed quietly or rejected entirely. Therefore our 'benchmark' was completely external to us, and in some cases intellectually adversarial.

"We chose to focus on some of the most well-audited, secure, and heavily tested pillars of the world's software ecosystem. Among them, OpenSSL stands out. Industry estimates suggest that at least two-thirds of the world's Internet traffic is encrypted using OpenSSL, and a single zero-day vulnerability discovered in it can define a security researcher's career. It is a very hard target in which to find real, valuable security issues.

"In late summer 2025, six months into starting our research, we tested our AI system against OpenSSL and found a number of real, previously unknown security issues. The Fall 2025 OpenSSL security release contained a total of four CVEs. Three of those four were discovered, responsibly disclosed, and in some cases even fixed by us (or more precisely by our AI system); and two were rated as moderate severity issues; and the third as low severity.

"For context on our approach: our system handles the full loop: scanning, analysis, triage, exploit construction (if needed and possible), patch generation, and patch verification." I hope you're listening to this, Leo, because this is astonishing. I mean, it has happened. "Humans choose targets and act as high-level pilots overseeing and

improving the system, but don't perform the vulnerability discovery. On high-profile targets, we additionally review the resulting fixes and disclosures manually to ensure quality, although this only rarely changes anything.

"Today, January 27th, 2026," meaning just last week, "OpenSSL announced a new security patch release, publishing 12 new zero-day vulnerabilities, including a very rare high-severity one. Of the 12 announced, we at AISLE discovered every single one of them using our AI system. Adding these new 12 to the three out of four CVEs we already had in 2025 previously, this means that AISLE, and by extension AI in general, is responsible for discovering 13 out of the 14 zero-day vulnerabilities in OpenSSL in 2025.

"Both the count and the relative proportion have been increasing as a function of time and are overall historically very atypical, with the most recent 12 vulnerabilities spanning a significant breadth of OpenSSL's codebase. Even a 'low' severity CVE is a higher bar than might be obvious. The vast majority of reported issues don't qualify as security vulnerabilities at all. Most are bugs that get fixed without CVEs as standard patch releases. To receive a CVE from OpenSSL, an issue must pass their conservative security posture and be deemed important enough to track formally.

"Low" severity in OpenSSL still means a real, externally validated security vulnerability in well-audited critical infrastructure. In five cases, AISLE's AI system directly proposed the patches that were accepted into the official release, following a human review from both AISLE and OpenSSL. Matt Caswell, Executive Director of the OpenSSL Foundation, said this about the findings: "Keeping widely deployed cryptography secure requires tight coordination between maintainers and researchers. We appreciate AISLE's responsible disclosures and the quality of their engagement across these issues."

Tomas Mraz, the CTO of OpenSSL, said about the newest security release the following: "One of the most important sources of the security of the OpenSSL Library and open source projects overall is independent research. This release is fixing 12 security issues, all disclosed to us by AISLE. We appreciate the high quality of the reports and their constructive collaboration with us throughout the remediation."

The researcher at AISLE continues: "The assigned CVEs still don't represent the full picture here. Some of the most valuable security work happens when vulnerabilities are caught before they ever ship, which is," writes the researcher, "my ultimate goal. Throughout 2025, AISLE's system identified several issues in OpenSSL's development branches and pull requests that were fixed before reaching any release.

"Our AI discovered a double-free in the OCSP implementation. It was caught and fixed before the vulnerable code ever appeared in a release. Our AI also found a use-after-free and a double-free in RSA's OAEP label handling. It found a crash in BIO_sendmmsg/recvmmsg with legacy callbacks, and our AI discovered a location where important private key file permissions were not being set by the OpenSSL 'req' command.

"This is the outcome we're eventually working towards - vulnerabilities prevented proactively, not only patched after deployment retroactively. The concentration of findings from a single research team, spanning this breadth of subsystems and vulnerability types, is historically unusual for OpenSSL and is in my view in large part due to our heavy use of AI."

So, okay. I wanted to put AISLE onto everyone's radar. They are at Aisle.com, and they're going to be worth keeping an eye on. What became clear to me in looking around their site is that their work on OpenSSL - you know, they also found a handful of true vulnerabilities, five of them, that resulted in CVEs in cURL - was just for the sake of working to perfect their process. Their actual business is not the improvement of open

source software. That was just a happy proof-of-concept development side-effect. For them, OpenSSL served as a perfect test bed, allowing them to further test their AI-assisted code analysis system and capability. They're going to be offering this capability for hire to the likes of Apple, Google, Microsoft, and others - until they're acquired by some big fish - as a means of enabling their customers to similarly find and fix previously undiscovered bugs.

Their "About" page says of them: "We're not chasing trends. We're solving the toughest problem in cybersecurity. AISLE was built by security leaders and AI scientists who've seen both the scale of the threat and the limits of today's tools firsthand. Our goal is not to improve vulnerability management. It's to end the backlog. To close the loop. We believe in something bold but measurable: zero exploitable vulnerabilities. Not as a slogan, but as an achievable outcome."

And explaining their mission, they wrote: "We started AISLE after seeing the same pattern again and again: Attackers move faster, and defenders are forced to catch up - with too many tools, too much manual work, and a backlog that never disappears." You know, think Microsoft. "We built AISLE to break that pattern, not by adding another dashboard, but by creating something fundamentally different: an end-to-end autonomous Cyber Reasoning System that finds, fixes, and verifies vulnerabilities at machine superhuman speed and scale. AISLE cuts remediation time from months or weeks to minutes and seconds, bringing us closer to a future where software defends itself: Built by engineers, accelerated by AI, and designed for reality."

So among AISLE's angel investors is a Chief Scientist at Google, the CPO for AI Experiences at Microsoft, the Co-founder and Chief Scientist at Hugging Face, and a Research Scientist at DeepMind. So they have the backing of industry professionals who have understood that this had to happen, that it was going to happen. You know, our listeners have heard me assert over and over that "code" should be totally understandable by a sufficiently capable AI. It really means something that AI managed to find 15 out of a total of 16 CVEs in a system of code as carefully composed, maintained, and scrutinized as OpenSSL. It's truly a big deal.

And as they said, the OpenSSL Project does not hand out CVEs readily. They don't want to. It's also found five new CVEs, as I mentioned, in cURL. You know, and they don't care they didn't get a bounty. They didn't do it for that. They did it to test their AI against open source software.

We've recently seen that AI has made surprising strides in the generation of code. And now it appears we're on the brink of being able to leverage the same power, the power of AI, to dramatically improve the quality of both the world's existing and its newly written code.

My final observation is that every step along the way of this AI revolution, what we've seen has occurred much faster than certainly I and many observers expected, and each new hurdle is easily being surmounted. You know, "Oh! One AI is insufficient because the problem requires a context window that's too large, causing the AI to start becoming confused? No problem. Just divide the too-large-a-task into separate individual smaller tasks and deploy a team of AIs, giving each of them a specific subset of the puzzle." What's happening is truly incredible.

Having seen and understood the significance of what AISLE has already accomplished, coupled with the speed at which all of this AI is evolving, I now believe that there's a very real possibility that many or most of us will live to see the day when software bugs are eliminated. That no longer seems like a far-fetched, faraway goal. I think that's the thing that AI is going to do. And it's going to change the world.

Leo: I completely - I couldn't agree more. And every day I see the evidence of that. One of the things that makes this really interesting with AISLE, they mention it, but I want to underscore it, is this stuff is incredibly fast, and it works all day, all night, tirelessly. So it, I mean, yeah, okay, maybe it's slower to find a bug than a human would be. I don't think it is, but let's say it is. It doesn't matter. You can assign 20 of them to do it, and they'll work all night for you and solve this problem.

Tomorrow we're going to interview a guy who's written some really interesting software called Gastown that's designed to be used with these AI coders like Claude Code. And what Gastown does is it creates, for every project, it creates roles. So you've got a refinery. You've got a crew. You've got something called polecats. You've got a witness. You have a mayor who runs the whole thing. You have a deacon.

Steve: You're talking a collection; right?

Leo: It's a collection.

Steve: That seems to be the next step.

Leo: And they work in concert, and they check each other. So if, you know, if somebody kind of goes off the rails, the mayor steps in and says, wait a minute. Until you see it at work, it's hard to believe; you know? I have a switch in my Claude Code that I turn on called Test-driven Development. I said, no, no, write a test for everything. And I don't want to hear from you until every test passes. Right? And it writes much more complete tests than I do. Because it doesn't have the biases that I do. It just says, well, I've got to test everything. There's also an overseer. That's the human. And it's just really - it's happening so fast, and so much creativity is being put into it. And there's a lot of risk. I acknowledge.

Steve: And I just shared with everybody that story in The Wall Street Journal saying that the software-heavy tech companies had just crashed.

Leo: There's a good reason.

Steve: Because people are realizing that, you know, hey, I don't need to pay Adobe. I just need to ask my AI to give me one.

Leo: Right. And honestly, I've talked about this before, we're in the age of hyper-personalized software. People can say, instead of saying "I have a photo-editing program, and I'm going to kind of interact with it to get it to do what I want," you just say what you want. And the computer writes a photo-editing program to do that for you. And it does it so fast. The other thing Harper Reed taught us very early on - I love Harper, he kind of lives in the future - is this stuff is so cheap in terms of, not dollars, but just generation, if something doesn't work, you throw it out, and you start over. You just go, oh, that's fine. It's disposable software. I wrote a tool that I only ran once that converted my Obsidian posts to the Day One Journal posts. And I

wrote it. I will never use it again. But it did it once, and it worked. And it did a beautiful job. And that's power software.

Steve: You know, and I'm sort of thinking about the brilliance of the way the Unix developers created Unix to be a whole...

Leo: Bingo.

Steve: Yes.

Leo: And so for instance my workflow with story processing is three different agents doing different things that pipe one to the other. That's exactly right. That's just why you asked me a few weeks ago if it's helpful to be a programmer or not. I think it is helpful to be able to think in that way in terms of processes and flow and planning.

Steve: But the job is changing.

Leo: But the way you do it, you don't write the code anymore, you write the prompt. But you still have to think in that kind of fashion, I think. Yeah, it's changing dramatically. I really want you to try it, just for fun. Get it to write assembly. I'm not kidding. It'll write assembly perfectly well. In fact, okay, here's a - here would be fun. Tell it, hey, this SpinRite program, it's in assembler. Here's the code. Can you make a Mac version? And just let it go. See what happens. I know. It's kind of - I know. It's a little weird and scary; isn't it. For somebody who's spent his life hand-coding software. Very much.

Steve: Who loves, who loves - it's not like labor. You know, like I always felt guilty when I said I'm going to go to work. It's like, it's not work. It's what I want to do.

Leo: It's like talking to a guy who carefully chisels a piece of furniture, and then saying, "See this thing called a lathe?"

Steve: Yeah. Here's a laser printer that can take that wood, yeah.

Leo: Right. And you know what, here's the good news. I love to code, too. And I will always code. I know you will always code. But now it's something we do for fun. Because we love it. Not for any other reason. Not because we have to.

Steve: Okay. Break time, and then a piece of errata, and then some feedback.

Leo: Now, that's what I love about you, Steve. Because you're very wide open to this. There's a lot of people who are saying, oh, no, you know, it's slop, it's slop. It's not - and you're very - you understand. This is transformational stuff that's happening.

Steve: I think we're going to see the end. We're going to be alive. You and I are, you know, I'm 70, you're almost.

Leo: I'll be there any minute.

Steve: I think we're going to - I don't think - I think five years from now we're going to, I mean, I think this AISLE company is going to get snapped up by one of the big fish in a heartbeat. I'm sure that's the way they profile themselves. And this is going to change software. Bugs are going to be over. Which doesn't mean the podcast is over because we still have the humans in the loop.

Leo: Yes.

Steve: And, you know, the title of our talk, our presentation next month at ThreatLocker is "The Call Is Coming From Inside the House."

Leo: I can't wait. That's going to be so much fun. All right.

Steve: Okay. So I have an important correction to share, thanks to one of our Irish listeners who took the time to explain an important nuance of Ireland's law-passing process. Our listener is James Pelow, who said: "Hi, Steve and Leo. Long-time listener and happy Club Twit member." He said: "I wanted to flag something from Episode 1062 regarding Ireland's Communications (Interception and Lawful Access) Bill." He said: "By way of background, I work as an Irish translator specializing in, amongst other things, Irish and EU legislation, translating to and from Irish, our national and first official language. I encounter these government press releases regularly, and I've translated a fair few of them myself down through the years, so I have some familiarity with how it all works.

"In this case, the bill has not actually been passed. In fact, it hasn't even been drafted yet. What the press release announces is that the Cabinet has approved the Minister's proposal to begin developing the legislation. Officials are now tasked with preparing a 'General Scheme' - which explains that term that we saw when I talked about this - "essentially a detailed policy outline which the Minister says he hopes to publish sometime in 2026. Only after that would the actual legal text be drafted, subjected to public consultation, scrutinized by parliamentary committee, introduced to the" - whatever that word is, O-I-R-E-A-C-H-T-A-S. I'm sure if you said that...

Leo: Irish, I'm sure.

Steve: ...with an Irish accent it would sound good.

Leo: Yeah, it makes more sense.

Steve: Anyway, he says, "(our parliament), debated in both the Dail (lower house) and Seanad" - sorry, James - "(upper house), and signed into law by the President, assuming she has no concerns about it being unconstitutional." In other words, boy, is it not a law.

Leo: Yeah, okay.

Steve: It has a long way to go.

Leo: It's a general scheme is what it is.

Steve: Yeah. It's an ambition, probably, better than anything else. So he said: "To clarify the terminology, in Irish law, a 'Bill' is proposed legislation still going through Parliament, while an 'Act' is legislation that has been passed and is in force." So the 1993 Act is current law. This proposed Bill isn't even a bill yet. It's an announcement of intent to draft one.

He said: "I'm personally of the view that this proposed legislation would actually be a significant improvement on the current regime. Under the 1993 Act, interception warrants are authorized by the Minister for Justice alone. No court order is required. Oversight is purely retrospective. A designated High Court judge reviews the operation of the Act and reports annually to the" - whatever that word is - "(prime minister), but doesn't approve individual" - James, if you can translate Irish into English, more power to you because this doesn't look like anything, Taoiseach.

Leo: I can play the pronunciation for you, if you want. I think I have it here. Let me see.

Steve: T-A-O-I, so we've got three vowels in a row. And then S-E-A-C-H.

Leo: You know what, none of the dictionaries, which normally have recorded pronunciations next to these words - oh, here we go. Here's one. None of them are playing it. You have to imagine it.

Steve: Anyway, the proposed...

Leo: Tea-shook.

DICTIONARY: Tea-shook.

Leo: Tea-shook.

Steve: Yeah, that's the Prime Minister.

Leo: Yeah. Tea-shook.

Steve: Anyway, so...

Leo: Tea-shook. Yeah, you know.

Steve: "The proposed bill would introduce a requirement for proper judicial authorization for the first time, along with an Independent Examiner and a formal complaints process covering all the powers." He says: "But to get a proper read on it, we'll have to wait for the actual text." And then he said something that I'm sure is thank you or goodbye or good luck or something. "Beir bua. James."

Leo: Yeah, I love Irish. I mean, I love it when they speak it. I couldn't pronounce it for the life of me.

Steve: Yeah.

Leo: That's wild. Okay. So we don't have to worry about it yet.

Steve: No. We do not have anything to worry about. You know, it sounds like from what James explained they're going to be - whatever they do will have way more formal controls on it than were in place in 1993. But they're still, I mean, it did say what we shared. It's not a law, but it's an intent. It's a wish for one, where they get to have access to anything. That's what they're saying. And remember, that was the one where they're saying we're going to give ourselves the right to install spyware on people's phones if we think that's what we need to do. So, well, I can't say "coming soon," but, you know...

Leo: Let's hope not.

Steve: It's the intent.

Leo: Yeah.

Steve: And we know that Germany expressed the same intent, which is what we'd also talked about last time. Okay.

Ronnie Morgan said: "I thought you'd be interested to know that Gemini recommended using your DNS benchmark. I have been working on changing DNS resolvers at work and was using Gemini to complete the task, mostly for fun. Before I flipped the switch, I asked Gemini what would be a good way to test the performance of the old resolvers versus the new ones, and its number one suggestion was your tool. Which I honestly didn't even think about until it suggested it. And the result? New servers are performing very well."

He signed off saying: "Thanks for a great tool. I'm looking forward to the surprise you have in store for the paid version of it," he says, "(which I've already purchased), and am looking forward to SpinRite 7.0." So thanks, Ronnie, for sharing that. I'm at work implementing some rather deep changes to GRC's eCommerce system, which I originally created and wrote in assembly language, with no help from AI, in 2003. And I haven't touched it since, in 23 years, and I'm excited about the system's forthcoming new features. But I'm going to continue to keep quiet about them until they're implemented,

tested, and ready, because people are going to immediately rush to it and want them, and I won't be ready.

So speaking of Gemini, I recently heard from Panos, the author and publisher of NuevoMailer, that marvelous emailing system I chose as the backend database and mail management platform for GRC's email list system, which all of our listeners are using. NuevoMailer is what mails nearly - I think it was 19,906 pieces of email on Sunday afternoon. None of them bounced back this time. This time Outlook and Hotmail had no complaints, whereas last week they bounced 1,500-plus, which I then later remailed with no complaints. So again, as you said, Leo, you know, anti-spam false-positive. Anyway, Panos has become a listener of this podcast, and he dropped me a note to share a chilling Gemini AI-related event that he suffered last week.

He wrote: "Hi, Steve. Listening to the latest podcast. And speaking of AI, here is what happened to me last week. The Gemini extension in Visual Studio Code, while in Agent mode, wiped out all files and folders from my project." He said: "I noticed it got into a loop doing/undoing the same changes in two files. So I stopped it. I switched to the File Explorer to pick something else, and it was empty. Not even in the Recycle bin. The response from Gemini?" I just love this, Leo: "That sounds incredibly frustrating."

Leo: Oh.

Steve: I'm sorry, Dave.

Leo: I'm sorry, Dave. I didn't mean to do that.

Steve: We're going to leave the airlocks closed, and you're going to suffocate. But so Gemini says: "That sounds incredibly frustrating, and definitely not the kind of 'assistance'" - it has in quotes, god - "not the kind of 'assistance' anyone wants from an AI. I'm sorry you're dealing with data loss. Recent reports, including documented issues in late 2025 and early 2026, have highlighted a bug where the Gemini CLI and VS Code extension can occasionally misinterpret conversational context as destructive terminal commands (like `rm -rf`)" - meaning, you know, recursively delete, you know, remove - "or fail during file-moving operations, causing files to vanish. Since these deletions often happen via the extension host, they might bypass your OS Recycle Bin, but VS Code has a hidden safety net that can often save you."

So then Panos continues: "The 'safety net' did not help much. It is only backups of recently edited files. Files that you've never opened with VS code are not there. Fortunately, I had another IDE which kept backups. Now I have a task running a bat file twice a day making backups of important projects to a different drive." Yeah.

So, you know, we've been exploring and promoting the idea of AI-driven software development, and we've seen instances where AI is aimed at an existing GitHub project and then takes over; you know. So I just wanted to share Panos's hair-raising adventure and suggest that anyone who might be similarly vulnerable running, you know, maybe run a separate, entirely disconnected project backup system, you know, something that the AI is not involved with in any way because, you know, belt and suspenders.

Leo: The way I handle this, and Panos, you should consider it, too, is everything's on GitHub, and you commit after every change. So there's always a way to pedal back.

Steve: To rewind.

Leo: Yeah. That's the beauty of a, you know, source repository like Git is you can always rewind. In fact, you frequently do want to rewind. Like, oh, that really screwed things up. Let's go back to the previous version.

Steve: I often - I will make what I call a "checkpoint." And then I'll go do something. And if I end up really tangled up, you know, like the first - the DNS Benchmark from 2008, because IP addresses fit in a 32-bit register, I mean, it was so difficult to switch it to 128-bit IPv6. And then strings, URLs. And so I kept going forward. I'd go, oh, and I'd rewind. I'd go back, and I'd start again, and each time I learned something that I hadn't anticipated.

Leo: Exactly.

Steve: Until finally I got it to work.

Leo: It's like a videogame. You save regularly so that if you run into something...

Steve: When you die you get resurrected. You get resurrected. You get resurrected, yes.

Leo: It's actually really handy. Almost all these tools use GitHub. And it's trivial to say, you know, to your agent, you know, every time we finish something, commit, you know, I always say - and it just does it automatically now - commit, push, and build. That's the other thing. I never used to use GitHub CICL process where it will build software.

Steve: Right.

Leo: It builds it now. And it builds it for multi-platforms. It builds it for three different platforms. So it's cross-platform. I just say, you know, okay, good. Good job. Commit and build. And then I go do something. We live in...

Steve: So it changes us from coders to project managers.

Leo: We're bosses, yeah.

Steve: Yeah.

Leo: Or the mayor, in the case...

Steve: So also being a listener, Panos also shared, he said: "P.S.: I attach something that happened to me today. Of course, I did not press Win+R." So he attached an example of one of the most terrifying social engineering hacks floating around today.

Leo: This would get a lot of people, I think.

Steve: It would. And that's my concern. You know, we've all encountered CAPTCHAs that we're asked to solve. That's a thing now. And more recently when we're attempting to visit a site that's hosted by Cloudflare, we'll encounter an intercept screen that asks us to wait a moment while it verifies that we're human. Sometimes that intercept will self-resolve, and other times we're asked to click on a checkbox to affirm our humanity. Presumably, since fancy JavaScript has been profiling our connection in some way, but it also wants to watch us as we servo the mouse over to the checkbox and click it.

So in a deviously brilliant social engineering hack that's obvious only in retrospect, bad guys realized that they could spoof the increasingly familiar Cloudflare intercept event and get people to follow additional innocuous-looking instructions. I know that a great many of us serve as the "computer experts" for our friends, neighbors, family members and fellow employees. So we've developed an appreciation for how little anyone really and truly understands the computers they're sitting in front of and using. That's what makes this particular social engineering attack so devastating. It will obviously have a high success rate.

A week or two ago we shared the experience of another listener of ours who, while visiting at his mom's house, began receiving money transfer acknowledgements on his phone. He ran home to discover that something called "Screen Connect" had activated, and somebody was controlling his machine remotely and using it to transfer his money elsewhere without his knowledge or permission. Naturally, he wondered how such malware might have landed inside his machine. In this case, being a savvy Security Now! listener, it's unlikely that he would have fallen for this particular hack, just as Panos did not. But unless somebody really understood what they were doing, this would look like an entirely reasonable request.

The solution is for Microsoft to get proactive here. Just as Cisco has needed to with their own networking gear, Microsoft needs to soberly recognize that Windows users are not expert users anymore. Less so every day. Over time, you know, they're becoming less expert. Clipboard? What's a clipboard? We see this recognition in many other areas of annoying preemptive handholding by Microsoft in Windows. I have two Windows machines that I don't care much about which are logged in with a Microsoft account. What a mistake - and lesson - that has turned out to be.

Microsoft is pushing everyone to login with a Microsoft account then they repeatedly brutalize anyone who does not. It's becoming annoying listening to Microsoft over and over and over again, you know, telling me that I need to turn on backups on my PC. It's like, no, I don't. Leave me alone. But I do a major update, and it's back to the, like, setting up Windows screen, you know, making me tell them like four times in a row, no, I am really sure I do not want you to do backups for me.

But in this case of what amounts to system clipboard abuse, which seems like a very serious problem that promises to wreak havoc, it would be trivial for Microsoft to track the source of any data that's placed onto the clipboard and take special measures when any clipboard data attempts to cross a security boundary. We know that today's web browsers are inherently high-risk containers and that a huge amount of effort has gone into browser containment.

A shared clipboard completely breaches browser containment. Right? Because it allows you to copy something from in the browser and paste it outside the browser. A shared clipboard is a fundamental weakness which just kind of crept up on us without anyone thinking about it. So the idea that it's possible for some malicious browser JavaScript, which originated from lord only knows where, to place malicious content onto the shared system clipboard and then instruct its user to execute that content by copying it into the Windows "Run" dialog - without having Windows raise a huge fuss with flashing lights and sirens and are you sure.

I mean, I'm sure I don't want backup. I am sure that I would like to have Windows warn me if something that I didn't manually put on the clipboard somehow got there and is about to be pasted into a run dialog. It seems to me it is the height of non-proactive irresponsibility on Microsoft's part. So Microsoft, if anyone there is listening, get this fixed because this problem is not going away. Burying your head in the sand is not going to fix this. You know, this is a problem.

Nick Mapsy said: "Hi, Steve. I just got to the part in last week's podcast where you break down how your ISP can snoop on you. You point out that once we're all using TLS 1.3, the most they can do is track what IP addresses you visit. But I want to point out that, even then, there's a much bigger privacy threat they can pose. As you said, ISPs know who you are and where you live. Third-party cookies can track where you've been on the Internet, but they don't inherently know who you are. ISPs can solve that problem for tracking companies. They could set up a marketplace where a company can ask, 'Who currently is at this IP address?' And the ISP would, for a price, tell them who you are, where you live, what's your email address, what's your phone number, et cetera.

"We already know that cell carriers have been selling real-time location data, so this is not a big leap at all. I haven't seen confirmation yet that this is being done, but I'm paranoid enough that this led me to finally start using a 24/7 VPN. I thought it might be worth pointing out on the podcast."

And, yikes! I have to agree with Nick's horrifying observation. Again, no one has any evidence or proof or belief that this is happening. But every ISP is aware of their subscriber's current public IP address. And it must be that law enforcement has been able to ask an ISP exactly who was using which of the ISP's block of public IPs at any given time. That would seem logical. So I agree with Nick that imagining ISPs might monetize that knowledge in real time is not a big leap. I suppose it would be one benefit of an ISP using carrier-grade NAT, which we've talked about before, where users don't get public IPs. They get a block of private IPs because the ISP themselves, just as users are behind a NAT router at home, the ISP is behind a carrier-grade NAT router and is issuing private IPs to its subscribers, in which case they're anonymized by that. So that would be one benefit of that.

But, you know, ISPs do know who we are. Who knows what the fine print says, whether they are actually able to disclose our real-time IP to anybody who asks, even not law enforcement, but for commercial purposes. I don't know.

And Leo, what I do know is that we have one final sponsor to introduce.

Leo: Steve can keep count. That's so impressive.

Steve: I miscounted one week.

Leo: I know.

Steve: I don't always count.

Leo: Yeah, let's take a break, final break. And then we will finish up with our story of the week.

Steve: Mongo's Too Easy.

Leo: Again, it has nothing to do with "Blazing Saddles." Do you remember Mongo from "Blazing Saddles"? Do you remember that? He was played by an ex-football player, what was his name? Alex Karras. He's a big guy. Mongo. And, you know, I wonder, though, if MongoDB could have been named after Mongo from "Blazing Saddles." I could see that. I mean, Python is named after "Monty Python"; right?

Steve: Okay.

Leo: Okay, Steve.

Steve: Even though we kicked off this year with a podcast titled "MongoBleed," and I resisted talking about it again so soon, the security research I just found was just too much fun and too interesting to pass up. The competition for today's main topic, as I said, was that one I already shared about AI finding flaws in OpenSSL. Which I agree with you, Leo. I mean, this is a game change for the software industry. And, you know, on the creation side we just saw the stock market, you know, punish companies that produce software because, oh, I can make my own now.

Okay. So get a load of this one. The following posting was made by the DarkNet Army, posted to the dark web at 2:00 a.m. last year on October 1st. So this is, you know, a dark web posting by the DarkNet Army. "What's up, hustlers? I've been using this secret method since 2019..."

Leo: What's up, hustlers?

Steve: What's up, hustlers?

Leo: What's up?

Steve: "...to pull in steady cash every day, but it's starting to get crowded now. Before this method gets completely burned out, I'm sharing it here so you can jump on it and make some serious money for yourself. This isn't some complicated tech-heavy process. You don't need to know coding, hacking, or anything technical. If you can copy, paste, and click, you're good to go. I'll guide you through every single step.

"So what are we actually doing? Here's the deal: There are websites out there where businesses store their important information (think customer records, orders, employee details, et cetera) in a digital storage system. This storage system is called a database. But here's the crazy part. Some businesses leave their databases completely

unprotected, wide open on the Internet. They don't set up passwords or any security, which means ANYONE like you can access them with just a browser. Once you're in, you can delete their data, wipe it all clean, then leave a ransom note telling them to pay bitcoin if they want their data back. Sounds wild; right? Stick with me, and I'll show you how easy it is to do this.

"Why are these databases exposed? Most businesses use a type of database called MongoDB because it's fast and easy to set up. They use a tool called Mongo Express to manage it, basically a control panel for their database. The problem? Many businesses are careless and leave their Mongo Express control panels exposed online with no passwords. This makes them perfect targets. You don't even need hacking tools to get in. Need help? If you're stuck or have questions, hit me up. DM me on the forum. Message me on Telegram." He provides addresses for all that.

And "Final words," he says. "This method is stupidly easy and works like magic, but it won't last forever. Businesses are slowly waking up and fixing their Mongo Express setups, so use this while you still can. Follow the steps outlined below to take action, and you can start earning \$600 a day."

That was actually posted to the dark web. For the hustlers. I titled today's podcast "Mongo's Too Easy" because MongoDB's continuing exploitation is now in the hands of the script kiddies. It turns out there's another "market" out there, such as it is, where there are no sophisticated intrusions with multi-terabyte exfiltrations of data, fancy command-and-control servers with dynamically rotating and changing time-based DNS domain lookups, encryptions and keys and all that. Nope. All of the data contained within exposed MongoDB instances are simply being deleted. In its place is a ransom note explaining that the data can be returned once payment of \$500 or \$600 in bitcoin has been received.

Just to be clear, that's not true. They don't know how to do any of that. They're script kiddies. Instead all of the databases' data was permanently deleted, and a bogus ransom note is being left behind. It's a bogus ransom note because payment of the ransom has no effect. None. No data is ever returned because it was irreversibly deleted from the database.

These are not the traditional serious attackers who hack, exfiltrate, encrypt, and extort. No. This is the bottom of the market. These attackers are trading on the "reputation," such as it is, that the high-end attackers carefully established long ago for honoring the payment of their extortion demands. Those guys are serious. These guys are not. The high-end attackers realize that, if they want their demands, which often run into the many millions of dollars, to be taken seriously, and paid, they need their victims to really believe that payment will result in the return of the stolen data and its subsequent deletion so that it never leaks publicly. If the high-end attackers do not honor their agreement upon the payment of ransoms, the high-end of this market will fail.

A Canadian cybersecurity firm known as "Flare Systems" posted a great piece from which I excerpted that earlier posting. The title of their posting last Monday was "MongoDB Ransom Isn't Back. It Never Left." They wrote: "Between 2017 and 2021, there was a series of research publications about MongoDB ransomware exploitation campaigns. These blogs described the same pattern. Someone in an organization made a mistake, which left MongoDB exposed to the world. The problem was that this MongoDB didn't require any special authorization or password. So anyone over the Internet could have accessed and controlled that database.

"Here's the sequence of events for attackers who abused these exposures: Threat actor finds a MongoDB database. They copy everything to their own device. They delete everything on the victim's computer. In place of the database, they leave a ransom note.

The ransom note claims: Pay hundreds of dollars in the next 48 hours or the database would be permanently deleted.

"That was five years ago. But since then, there have only been a few stories about ransom against MongoDB. However, a couple of months ago, we conducted a pentesting exercise for a small to medium-size business. The organization had 12 MongoDB instances, and two of them were exposed to the Internet with a ransom note inside. Reminding us of the MongoDB ransom campaigns, we decided to create and run a honeypot exposing MongoDB secrets." And Leo, if the Thinkst Canary can support a MongoDB, that might be a fun thing for people to play with.

Leo: Oh, that's a good idea. I have to check and see if I can do that. I bet I can. That's great.

Steve: Yeah. So they said: "A short Google search indicated that under the surface there are many similar stories. One story reflects the threat from a victim's perspective talking about a rising star tech startup that heavily relied on MongoDB as a database being hacked and extorted for \$25,000. In this blog we analyze the current MongoDB ransomware threat.

"MongoDB ransom attacks are not driven by advanced exploits or novel malware. They are the predictable outcome of Internet-exposed, unauthenticated databases. As long as insecure deployment patterns continue to propagate through tutorials, container images, and copy-paste infrastructure, these attacks will remain cheap, scalable, and profitable for threat actors, and costly for organizations without proper controls.

"The MongoDB ransom ecosystem demonstrates that real risk often emerges from the intersection of deployment patterns, configuration shortcuts, and attacker monetization models, rather than from advanced exploits alone. The attacks exploit MongoDB databases that are exposed to the Internet with default, unsecured configurations - no password, open ports, and so on."

Leo: Oh, yeah. Here, I can turn on MongoDB.

Steve: Very cool.

Leo: Let me just tell you it's on port 27017, in case anybody wants to get in.

Steve: Yeah, that is the default MongoDB port.

Leo: Yeah, no problem. So I've got a Windows server running with a MongoDB on it. Just come on in. Hack away. You're more than welcome. I'll know immediately, though.

Steve: Very nice.

Leo: Isn't that great that I can make that Thinkst Canary be that? That's fantastic.

Steve: Very cool.

Leo: Sorry. Go ahead.

Steve: So they said: "Automated scripts (bots) scan for vulnerable instances. Once an open database is found, the data is typically exported, or simply deleted. The collections are dropped, and the new collection containing a ransom note is inserted. Threat actors demand payment in bitcoin (often around 0.005 BTC, equivalent today to between \$500-600." Actually, that depends upon when you look. Bitcoin's been having a little rough time of it lately. They said: "...to a specified wallet address, promising to restore the data. However, there is no guarantee the attackers have the data, or will provide a working decryption key if paid. These incidents, sometimes referred to as the MongoDB 'Apocalypse,' affected tens of thousands of servers.

"Victims who have paid the ransom often reported receiving nothing in return, or finding the provided data/keys were useless, leading to permanent data loss. Thus, security experts strongly advise against paying the ransoms." On the other hand, 500 or 600 bucks. Part of the key to this working at all is that the bad guys put no effort into this. A bot found it. A bot dealt with it. And they're not asking for millions of dollars, they're asking for five or 600 bucks.

They said: "We set up a MongoDB honeypot" - and so has Leo - "on a container infrastructure, connected to the world without authentication. We deployed the container in various geolocations. It didn't take long; a few days after we set up the containers, we saw the ransom note in all the servers."

They then show the MongoDB shell running and the command "show dbs" - you know, databases - which results in the listing of a file titled "READ_ME_TO_RECOVER_YOUR_DATA." After using the MongoDB shell to switch to that file, it is dumped to the console. And it reads: "All your data is backed up. You must pay 0.0054 BTC to" - and then a bitcoin address. "In 48 hours your data will be publicly disclosed and deleted. For more information go to" - and then they have a website address, 2info.win/mdb. They said: "After paying, send mail to us," and then they have an email address, "rambler+1Y08BU@onionmail.org," they said, "and we will provide a link for you to download your data. Your DBCODE is" - and then the same token - "1Y08BU."

So they said: "We observed this attack. We started collecting threat intelligence to better assess this threat and associated risks. We found hundreds of relevant results including this MongoDB ransom tutorial." The one that I showed you guys above. That's the note that I showed before. Then, under the heading "Why and How Does the MongoDB Attack Technically Happen?," they said: "MongoDB is a widely used NoSQL document database designed for flexibility, scalability, and speed. Instead of rigid tables and schemas, MongoDB stores data as JSON-like documents, making it a natural fit for modern applications that evolve quickly and handle diverse data types.

"It is commonly used in web and mobile applications, SaaS platforms, IoT backends, real-time analytics, content management systems, and microservices architectures. Its ability to scale horizontally, replicate data across nodes, and support high-throughput workloads has made MongoDB a popular choice among startups and enterprises alike, and particularly in cloud-native environments where agility and rapid development are key. With this understanding, we leveraged Flare" - which is a tool that they use, their own in-house tool - "to identify publicly shared code snippets that explicitly configure MongoDB servers to be exposed to the Internet without authentication.

"This approach is based on the assumption (validated repeatedly in real-world incidents) that individuals and organizations often rely on ready-made Docker images and copy-paste configurations from Docker Hub and GitHub when deploying infrastructure. Using Flare, we searched for code artifacts containing the command pattern that would bind MongoDB to all network interfaces and enable unauthenticated access by default." And they give a sample of such a string in their posting.

They said: "This configuration results in a MongoDB instance running inside a container that accepts connections from any IP address. When the container port is bound to the host and exposed externally, any Internet-originating traffic can connect directly to the database. In their default configuration, these MongoDB deployments do not enforce authentication" - again, "In their default configuration, these MongoDB deployments do not enforce authentication or require credentials, allowing unrestricted access to any party that can reach the service. As a result, this code pattern leads to publicly exposed MongoDB instances.

"Over a three-month analysis period in our query, we identified 763" - okay. So 90 days, three months, analysis. They said: "We identified 763 container images uploaded to Docker Hub containing exactly this insecure configuration. These 763 container images spanned 30 distinct namespaces. Most of these images appear to be intended for personal or experimental use and have only a few hundred pulls. However, we also identified two widely used projects with more than 15,000 pulls each that included the same insecure setup." Okay. So Docker Hub is hosting two specific images for which 30,000 deployments have been made insecurely.

They said: "While these numbers alone do not appear significant, this represents only one of the many common ways MongoDB is inadvertently exposed. We highlight this pattern to illustrate how easily insecure configurations propagate, and how widespread such exposure can become. Out of curiosity," they said, "we also searched for some exposed credentials. We found 17,909 potential results for a specific user:password exposure (one of many potential search terms). Out of those, we found at least half of them as valid credentials that can be abused by attackers.

"The diversity of sources illustrate the low level of password hygiene in the wild and how easy it is for attackers to obtain credentials in the wild. We found exposed credentials in coding repositories and registries (such as GitHub and Docker Hub), dark web forums, paste sites, and Shai Hulud victims. We used Shodan to identify Internet-connected MongoDB services. Our analysis revealed more than 200,000 servers running MongoDB that were publicly discoverable." Again, remember, there are very few instances where you actually need public exposure from MongoDB database. It is meant for internal infrastructure, not remote access. "200,000 servers they found running MongoDB that were publicly discoverable. Of these," they said, "slightly over 100,000 instances disclosed operational information, and 3,100 were fully exposed to the Internet without access restrictions.

"Among the 3,100 fully exposed servers, 1,416," that is to say 1,416 instances, "had already been compromised, with their databases wiped and replaced with a ransom note. In nearly all cases, the ransom demand was approximately \$500 U.S. in Bitcoin. Notably, only five distinct Bitcoin wallets were observed across all incidents, with the wallet associated with the ransom notes left on our servers appearing in over 98% of cases." In other words, one attacker is out there. Basically their business model is just scanning the Internet for morons who put data on a MongoDB. And, you know, they delete it and put a ransom note up and hope to get paid. 98% of all of the ransom notes they've seen pointed to the same bitcoin wallet. They said: "This strongly suggests the activity is attributable to a single dominant actor, likely the same attacker documented in our previous dark web research.

"The data reveals an interesting discrepancy." They said: "While Shodan identified 3,100 servers as fully exposed to the Internet, our analysis shows that only slightly less than half of these instances were actually found to be compromised and wiped. Based on the Shodan data we found, a little more than 95,000 of the more than 200,000 exposed servers had at least one vulnerability." So there are also these servers are vulnerable in addition.

So under their "Prevention and Mitigation" section, they enumerate all of the expected steps and measures. You know, avoid exposing MongoDB directly to the Internet. Enable authentication and authorization. Restrict network access. You know I'm a big fan of IP address filtering. Why let the world have it? Why expose it to Asia, for example? If you have to have it exposed in the U.S., then do some geolocating. That's no longer difficult to do. They say: "Harden container and cloud deployments. Implement continuous exposure monitoring. Isolate the database. Audit access logs. Assess data integrity. And patch and upgrade." Right. So, you know, all of that amounts to standard and expected best practices. Don't expose the darn thing to the Internet. Period. Why? You know, exercise any sort of security hygiene.

So anyway, my two final points are: The first is one of my primary - we wake up and smell the coffee. It's not that it's impossible for authentication to work, it's that it absolutely must not be relied upon to work. It should never be the only thing standing between attackers and disaster. It should only ever be one of multiple lines of defense. One of my favorite things that I hit upon last year, thanks to this podcast, is the observation that the ONLY servers that should EVER be exposed to the Internet are those that are meant to be accessed anonymously by everyone. In other words, no authentication on purpose. No authentication by design. Things like web servers and email servers and DNS servers that everyone is expected to access. Their job is to provide anyone who comes knocking a connection and access.

This means that NOTHING that requires a logon before its services can be used from the public Internet should ever be widely exposed. I know it sounds nutty and impractical. But almost all systems and services could be set up that way if their IT people cared to do so. Pointing fingers at Microsoft, Cisco, or whomever after the fact, and blaming them for their authentication failures may shift the blame. But a more robust overall network design could have prevented their failure from also highlighting yours.

And I said I had two points to make. The second point flows from this line of Flare System's conclusion. They write: "Attackers did not rely on sophisticated exploits or zero-days. Instead, they abused insecure defaults." This further supports the pessimistic contention I ended with last week. AI may help us to find flaws in our software. Now we know that's almost certain to happen. Yay team! That's great! But unfortunately, while AI may be getting smarter, it also shows no signs, nor hope, of being able to make us humans any less dumb. AI won't fix what amounts to laziness and lack of attention to critically important details, configuration mistakes, and default setups. That's on us. There's just no excuse for MongoDB, for example, to still, as we enter 2026, be in the sad state it is. It's truly unconscionable.

Leo: Well, maybe they'll listen to this show and figure it all out, Steve. Certainly I have. Now I have to go open all the ports on my router so that my AI assistant can do everything.

Steve: Oh. That's - that's good. Give it your credit card number. Give it your family history.

Leo: You only live once, Steve.

Steve: YOLO, baby.

Leo: YOLO, baby. It's so tempting. You know, I'm sitting here looking at - I'm giving it OAuth credentials to my Gmail and my Google Drive. Well, how else is it supposed to triage my email and upload files and...

Steve: And know what you're thinking.

Leo: And know what I'm thinking.

Steve: Yeah.

Leo: I already kind of gave it a brain dump. I'm going to also give it my Obsidian and Day One Journals, and it can know everything deep down in my inner secrets. But I'm not giving it my GitHub keys. No way. Steve Gibson is at GRC.com, the Gibson Research Corporation.

Copyright (c) 2014 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details:
<http://creativecommons.org/licenses/by-nc-sa/2.5/>